

本とハイパーテキストを融合したグループ指向作業環境の実現

市村 哲 前田 典彦 工藤 正人 松下 温

慶應義塾大学理工学部

本稿では、迷子の問題の解消と、ユーザの知覚上の負担の軽減を目指した共有ハイパーメディアシステムOpenBookを提案する。本メディアとハイパーメディアを統合することを考え、ハイパーメディアのブラウザの構築に本メタファを導入した。本形式のブラウジング手法は大量の情報にざっと目を通すことができるという利点を持っており、読者は本の内容の大筋を素早く把握し、要点だけを拾い出すことができる。さらに、問い合わせ検索機構とISMノード構造化法を組み合わせることでシステムに導入することによって、情報の管理形態としてハイパーメディア形態を用いつつ、本メディアの形態でその情報へアクセスすることを可能にした。

A Shared Hypermedia Supporting a Page-Leafing Style Browser

Satoshi Ichimura, Fumihiko Maeda, Masato Kudo, Yutaka Matsushita

Dep. of Science and Technology, Keio University
3-14-1, Hiyoshi, Kohoku-ku, Yokohama, JAPAN
E-mail: ichimura@myo.inst.keio.ac.jp

The OpenBook system using the book metaphor has the potential to prevent disorientation and reduce cognitive overhead. The system allows a user to leaf through a set of nodes retrieved from a hypermedia just like a book. Leafing through an electronic book takes advantage of a cognitive capability to skim or perceive the outlines of the contents. Moreover, the system supports a query-based access mechanism to facilitate global data access. Hence, hypermedia network is used for an underlying document storage space; a book metaphor-based browsing environment is for an easy-to-use user interface.

1 はじめに

データ管理形態の面から見たハイパーメディアの特色は、非定型情報をノードの中に埋め込み、非線形リンクを用いて複雑かつ自由にそれらのノードを結び付けて管理できることにある。また、グループウェアの基盤技術として見た場合、協同作業の作業対象となる非定型かつ複雑な構造を持ったデータを従来のデータベースで管理することは難しく、自由度に富んだハイパーメディア形式のデータ管理形態の方が適していることが多い。さらにユーザインタフェースの面から見た場合、リンクをたどることによりノードからノードへと情報を読み進めゆくことができるという機能（以後、ナビゲーション機能と呼ぶ）は、ハイパーメディアの定義とも言える重要な特徴であり、特に、定型的な構造を持たない（非構造）情報を検索するための技術として注目されている。しかしながら、ハイパーメディアの提供するナビゲーション機能はそれ単独で十分なデータアクセス手法にはなり得ない。たとえば、複雑なハイパーメディアネットワークの中でユーザが自分の現在位置を見失ってしまうという迷子の問題は、ハイパーメディアの抱えている代表的かつ本質的な問題である。すなわち、ハイパーメディアのデータ管理形態は他のメディアに比べてかなり自由度に富んでいるが、反面、いざユーザがその複雑に管理されたデータにアクセスしようとする、結局一つ一つシーケンシャルにノードをたどることしかできず、自由度に乏しい検索しかできないのである。

一方、本では、データの管理形態は線形であるが、ユーザが情報にアクセスする際には拾い読みができるなど自由度は大きく、実質上ランダムに情報にアクセスすることが可能である。これらのことから、著者らは、「OpenBook」ハイパーメディアシステムを設計するにあたって、情報の管理形態としてハイパーメディア形態を用い、その情報へのアクセス形態として本メディアの形態を導入した。

OpenBook は大規模ハイパーメディアシステムあるいは共有ハイパーメディアシステムとして設計されている。本稿では主に、OpenBook システムの設計理念、関連研究との比較、実現のための工夫点、実装されたプロトタイプシステムのユーザインタフェースに関して議論する。

2 システムの特徴と設計理念

著者らが以前開発したチームウェアデータベース [1]（共用データベースと個人用データベースをシステム内で統合し、さらにハイパーメディアの機能を持つデータベースシステム）の使用実験評価、および、市販のハイパーメディアシステムの使用評価を行った結果、ナビゲーション機能の抱える迷子の問題（disorientation [2]）と、ユーザの知覚上の負担（cognitive overhead [2]）の

問題が深刻であることを実感した。これらの問題のために、ハイパーメディアシステムを使用している時に、ユーザは「混乱」あるいは「嫌気」を感じてしまうのである [2]。そこで、OpenBook を設計するにあたっては、特に、マンマシンインタラクションを円滑にすることに重点をおいた。

2.1 本メタファの使用

OpenBook システムはユーザインタフェース部に本メタファを用いており、従来のナビゲーション機能のみならず本の持つ特徴を有効に活用した情報アクセス機能を提供している。たとえば、ユーザは OpenBook 上でハイパーメディア中のノード群を、本のページをバラバラとめくるように読み進めて行くことができる。本をバラバラとめくって情報を拾い読みしているような時、読者は本の各ページに書かれた詳しい記述を細かに読んでいる訳ではなく、むしろ、印象的な情報（たとえば、章のタイトル、図、表、太文字）だけに注意をはらっている。ハイパーメディアが提供する従来のナビゲーション手法に比べ、この種のブラウジング手法は大量の情報にざっと目を通すことができるという利点を持っており、読者は本の内容の大筋を素早く把握し要点だけを拾い出すことができる。一方、従来のハイパーメディアシステムは、「次々と現れるノード各々に対して、その中の詳細な情報を理解した上で次に進むべき方向を毎回選択しなければならない」というプロセスを義務としてユーザに課している。これは、次に進むためには一つ一つのノードの内容を立ち止まって熟視しなければならないという極めて煩わしい作業を読者に強制していることになり、ハイパーメディアのユーザに嫌気を感じさせる最も大きな原因の一つである。

さらにくわえて、本というメディアは我々にとって非常になじみ深いものであり、本の物理的性質がもたらす数々の利点により、読者は情報を読み進めて行く上で大きな安心感を得ることができる。たとえば、本のページの厚みによってユーザは自分の現在読んでいる位置を把握できたり、残りページ数を見ることによって、本に含まれている情報を読み終わるまでにどれだけ時間がかかるのかを大まかに把握することができる。また、ページをめくる動作によって本のはじめの方に向かってページをめくっているのか終わりに向かってめくっているのかをあえて意識する必要なく知ることができる。たとえば読者が本の中での自分の位置を見失ったとしても、本の初めからあるいは以前読んだ特定のページから再び読み始めれば、記憶に残っているページに必ずたどりつけることが保証されている。一方、従来のハイパーメディアにおいては、印象に残っているノードに再びたどりつけるという保証はなく、たどるべきリンクを一つ間違えただけで簡単に現在位置を見失ってしまうことにもなりかねない。人間にとって、再刺激によって印象的な情報を思い出すことに比べ、たどってきた道を思い出すことはは

るかに難しいのである。

くわえて、一般的なハイパーメディアシステムは多数のノードを画面上に散在させて表示する方法をとっているが、画面サイズが制約されていることと画面上が煩雑になるという欠点によって、ユーザにとって使い易い表示方法とは言えず [3]、本メタファの導入はこの問題の解決策にもなり得る。

2.2 問い合わせ検索機構

高速ページめくり形式のブラウジング機構に加え、OpenBook は問い合わせ検索機構を備えている。ハイパーメディアを大規模データベースとして見た場合、問い合わせ検索機構（インデックス検索機構、属性値検索機構、フルテキスト検索機構）は、必要な情報に直接たどり着くための手段として必須である [4]。特に、他人によってどのようなノードが追加されたか、どのようなリンクが付けられたかがわからない共有ハイパーメディアの場合、ナビゲーション機能と問い合わせ検索機構の融合が強求められる。

現在のところ、試作したプロトタイプシステムはインデックス検索機構をユーザに提供している。ユーザが、インデックス検索機構を用いて問い合わせを OpenBook システムに対して行くと、システムは条件に一致したノードの束を返答する。ただし、OpenBook システムの提供する問い合わせ検索は、従来のハイパーメディアに備わっている単なるインデックス検索とは異なり、ハイパーメディア中のリンク構造を加味した検索機構である。すなわち、ノード間の意味的つながり（文脈）を表現しているリンクを用い、潜在的に有益な情報も併せてハイパーメディアの中から検索するのである（以後、この検索法をストラクチャサーチと呼ぶ）。

その後、抽出されたノード群は「ISM ノード構造化法」によって線形になるように順序づけられ、ワークステーションの画面上に一冊の本の形態で表示される [5]。この時抽出された複数のノードは、多くの場合リンクによってお互いに結びつきあっているが、著者らが考案した ISM ノード構造化法に基づいて、このハイパーメディアの複雑なリンク構造はシステムによって線形に順序づけされる。

3 関連研究との対比

「自分がどこにいるのか、次にどこへ進めるのか、どうすれば必要とする情報にたどり着けるのか」、これらがハイパーメディアのナビゲーション機能を利用しているユーザが訴える迷子の問題である [2]。また、うっかり本来の目的を忘れて、興味をそそられたノードへとわき道に逸れたような場合、あるいは、目的の情報にたどり着いていない段階で、他の作業が割り込んできた場合、本来の検索作業を途中から続行することが非常に難

しいという問題もある [3]。さらに、前述したように、次々と現れるノード各々に対して、ノード内の詳細な情報を理解した上で次に進むべき方向を毎回選択しなければならないというナビゲーション操作におけるユーザへの負担は、ユーザに嫌気を感じさせる最大原因の一つになっている。これら、迷子の問題と知覚上の負担に対して、今までにも多くの研究者らが様々な対処策を提案してきた。

多くのハイパーメディアシステムが試みてきた方法の一つに、ノードとリンクの関係の概観図をグラフィカルに表示する方法がある。これらは一般に、ノードのタイトルをアイコンラベルとしてマップ上に適当に配置し、ノード間のリンクを線で表現している。たとえば、グローバル/ローカルマップおよびヒストリーパスを表示する Intermedia の Web View [2] や、リンクに属性を持たせた上でその属性を特定して概観図を描く NoteCards [6] と Neptune [7] のマップ表示機能などがその例である。しかし、リンクのタイトルから内容を推測できない場合も多く（オペレーティングシステムにおいて、ファイル名からファイル本体の内容がしばしば推測できないという状況に似ている）、ノードの内容をユーザに見せずリンクのタイトルだけを表示するこれらの手法は、必ずしもユーザにとって使いやすいものとは言えない。特に複数人で共有されるシステムの場合、ノードのタイトルを書いた人（執筆者）とそのアイコンラベルを見る人（読者）の間に解釈の違いが生じることも多く、実際に内容を確かめる必要性が高まる。さらに、ノンブラングラフの存在や画面サイズの制約によって、完全なグラフィカルマップを表示することは不可能である [3]。

一方、ナビゲーション機能と問い合わせ検索機構を融合しようとするアプローチも数々行われてきた（動的リンク機能を備えたシステムもこの範疇に含まれる）。これらのシステムに共通の主張は、データベースから検索されたノード中に存在するリンクをたどることにより、検索によって抽出されたデータ以外の範囲の情報にも即座にアクセスできるということである（従来のデータベースや電子ファイルシステムでは、検索されたデータに満足できなかった場合、条件を修正して一から検索をやり直すしかなかった）。たとえば、知識ベースに対してこの手法を適応した I³R [8] や、医療用ハイパーメディアとして開発された Medical Handbook システム [4] などがその例である。しかしながら、これらのシステムはナビゲーション機能を抱えている問題点を継承しており、ユーザの知覚上の負担 (cognitive overhead) を軽減していない。すなわち、データベースから抽出された複数のノード各々に対して、その中の詳細な情報を理解した上で次に進むべき方向をいちいち選択しなければならないというユーザの負担は軽減されておらず、ノードの内容を立ち止まって熟視しなければならないという煩わしい作業をユーザに強制している。

他のアプローチとして、Watters [9] らは transient

hypergraph モデルを提唱し、ノード間の仮想リンクや複合ノードを問い合わせによって一時的に生成するという方法を考案した。ユーザは、問い合わせによって生成された仮想リンクをたどることができるので、ハイパーメディアのような情報のブラウジングが可能なのである。仮想リンクの概念は ZOG [10] でも用いられていたが、Halasz [6] も述べているように、ユーザの手によって直接付けられた静的リンクの代わりに、問い合わせベースの仮想リンクですべて置き換えることはできない。すなわち、静的リンクをまったく許さないという仕様では、本来ハイパーメディアリンクが持つ「ユーザが直感的・個人的な思いつきで任意のノード間にリンクを自由に作成できる」という利点を損なうことになる。

4 OpenBook システムの使用方法

OpenBook システムは、情報の管理形態としてハイパーメディアネットワークを用い、情報へのアクセス形態として本メディアの形態を導入した、共有ハイパーメディアシステムである。この章では、著者らが開発したプロトタイプシステムの使用法および概要について述べる。

ハイパーメディアへのデータの蓄積: 複数人によって持ち寄られた情報はハイパーメディアの中に次々と蓄積されてゆく。ユーザがノードを追加する場合、そのノードに一つあるいは複数のキーワード（インデックスとして用いられる）を付けるか、そのノードとハイパーメディアの中に既に存在している任意のノードとの間に静的リンクを張るかした後、データをハイパーメディアの中に蓄積することになる。追加するノードにキーワード付けをする際には、後述の OpenBook マネージャによって支援され、また、付けられたキーワードは、ノードを管理しているデータベースとは別のリレーショナルデータベース（後述）によって管理される。

ハイパーメディアからのデータの抽出: ハイパーメディア中に蓄積されているデータはすべてのユーザからアクセス可能であり、誰でも情報を問い合わせ検索あるいはナビゲーション検索によって抽出することができる。ユーザが、問い合わせ検索機構を用いて問い合わせを行うと、システムは条件に一致したノードの束を返答する。この検索時、ハイパーメディア中のリンク構造を加味し（後述のストラクチャサーチの節を参照）、潜在的に有益な情報も併せてハイパーメディアの中から抽出する。その後抽出されたノード群は「ISM ノード構造化法」（後述）によって線形になるように順序づけられ、画面上に一冊の本の形態で表示される。

ページめくり形式のブラウジング: ユーザは本の形態で表示されたノード群を、高速ページめくり機能を利用

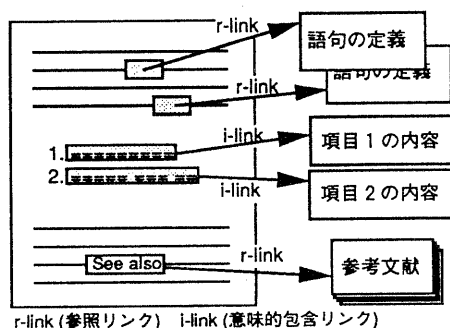


図 1: 参照リンクと意味的の包含リンク

して、あたかも本をバラバラとめくって情報を拾い読みしているようにブラウジングする。また、ブラウジングしている間に興味のある事柄（図、表、太文字など）が見つかった場合には、本のそのページに書かれた詳しい記述を読んだり、そのページの中に存在するリンクをたどってより詳しい情報を求めてナビゲーション検索を行うこともできる。

5 システム実現のための工夫

5.1 意味的つながりを表すリンク

Halasz [6] は従来のハイパーメディアのリンクを大きく参照リンク (reference link) と包含リンク (inclusion link) の 2 つに分けて、その意味的解釈 (semantics) の違いを説明し、区別することの必要性を示唆している。また、NoteCards [6] と Neptune [7] でもリンクに属性を持たせており、その意義が強調されている。くわえて、Walker [11] はドキュメント開発のためのハイパーメディアシステムを構築し、その中で包含リンクと参照リンクを効果的に使い分けている。

OpenBook システムでは、一般的な参照リンクと意味的つながり（文脈）を表現するリンクとを区別して管理しており、それぞれ参照リンク (r-link)、意味的包含リンク (i-link) と呼ばれる。図 1 を用いて、OpenBook システムにおける参照リンクと意味的包含リンクの違いを説明する。参照リンクは、参照文献を指したり、語句の定義が書いてあるノードを指したりするためのリンクであり、2 つのノードの間に粗な関係（それぞれのノードは本質的に意味的独立性を保っている）があることを示すリンクである。一方、意味的包含リンクは、指されている方のノードが、文脈から見て、指している方のノードに含まれる可能性があることを表すリンクである（この分類方法は Halasz や Walker らが行った分類方法と同一であり、このように分類することの実用性および有効性は実証されている）。

以上に参照リンクと意味的包含リンクの意味的な違

いを述べたが、さらに OpenBook システムにおいては、2種類のリンクのうち意味的包含リンクだけが問い合わせ検索処理時にシステムによって参照される、という機構上の違いがある。つまり、ユーザが、問い合わせ検索機構を用いて問い合わせを行うと、システムは条件に一致したノードの束を返答するが、この時、意味的包含リンクを加味し、潜在的に有益な情報も併せてハイパーメディアの中から抽出するのである（図2参照）。

5.2 インデックス/ストラクチャサーチ

OpenBook システムの提供する問い合わせ検索は2つのフェーズを持っており、それぞれインデックスサーチ、ストラクチャサーチと呼ばれる。以下これらについて述べる。

インデックスサーチを説明するために、「インデックススペース (index space)」と「ドキュメントスペース (document space)」という語句を文献[4]から引用する（この論文で、Frisse は様々なハイパーメディアのアーキテクチャに関して解説している）。OpenBook システムは階層型インデックススペースとネットワーク型ドキュメントスペースを組み合わせた形態をとっている（図2参照）。OpenBook システムの階層型インデックススペースでは、インデックスは階層化されており、IS-A 関係（継承関係）を保っている。たとえば、上位インデックスを指定して問い合わせ検索を行うと、そのインデックスの下位のインデックスを付けられたノードも同時に検索される。

インデックスサーチが終了した後、システムは続いてストラクチャサーチを行う。このストラクチャサーチは、意味的包含リンクを加味し、潜在的に有益な情報をハイパーメディアの中から検索するための機構である。もしあるノードが、インデックスサーチによって検索された任意のノードと意味的包含リンクによって結ばれていたならば、そのノードはストラクチャサーチの際に検索される。図2にも表されているように、「インデックスサーチによって検索された任意のノードを意味的に含むノード」と「インデックスサーチによって検索された任意のノードに意味的に含まれるノード」の両方が抽出される。

たとえば、図2に表されるインデックススペースにおいて、ユーザが問い合わせキーワードとして $i2$ を指定し、検索を開始したとする。するとシステムは、 $i2, i21, i22$ のいずれかのキーワードが付けられているノードをインデックスサーチによって検索する。 $\{N3, N5\}$ がインデックスサーチによって検索されたノードである。その後システムは、ストラクチャサーチによってノード $\{N1, N2, N3, N4, N5\}$ をハイパーメディアから抽出する。

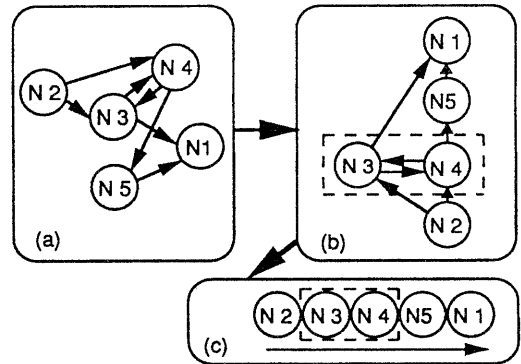


図3: 有向グラフ表示されたノードとリンク

5.3 ネットワーク構造の線形化

インデックスサーチおよびストラクチャサーチを経て検索されたノード群は、その後、線形になるように（本の形態で表すことができるように）順序づけされる。検索されたノード群は、通常、図2が示すように意味的包含リンクによって複雑に結びつき合っているが、システムは「ISM ノード構造化法」に基づいてネットワーク構造を持ったノード群を線形に順次並べて行く。ISM 法は社会システム工学の分野において開発された方式で、複雑な構造を定性的に分析して体系的把握するための一手法であるが[12]、著者らはこの ISM 法をノードを線形化するために用い、ISM ノード構造化法と名付けた[5]。

ネットワーク構造の線形化操作は大きく3つのフェーズ（可到達マトリクスの導出、階層ネットワーク構造の生成、階層ネットワーク構造の線形化）からなっている。以下にそれぞれについて例を用いて説明する（一部の記述を文献[12]から引用している）。

可到達マトリクスの導出: 図2を用いて、例えば5個のノード $\{N1, N2, N3, N4, N5\}$ からなるネットワーク構造を考える。このときネットワーク構造は図3(a)に示すような有向グラフとして表現することができ、これに対応した2値マトリクス A は式1に示すように表わされる（ノード間の直接的リンクの“あり”、“なし”を、“1”、“0”でそれぞれ表している）。式1はグラフ理論においては隣接マトリクスと呼ばれている。

$$A = \begin{matrix} & \begin{matrix} N_1 & N_2 & N_3 & N_4 & N_5 \end{matrix} \\ \begin{matrix} N_1 \\ N_2 \\ N_3 \\ N_4 \\ N_5 \end{matrix} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix} \quad (1)$$

こうして得られた式1の隣接マトリクス A に対して

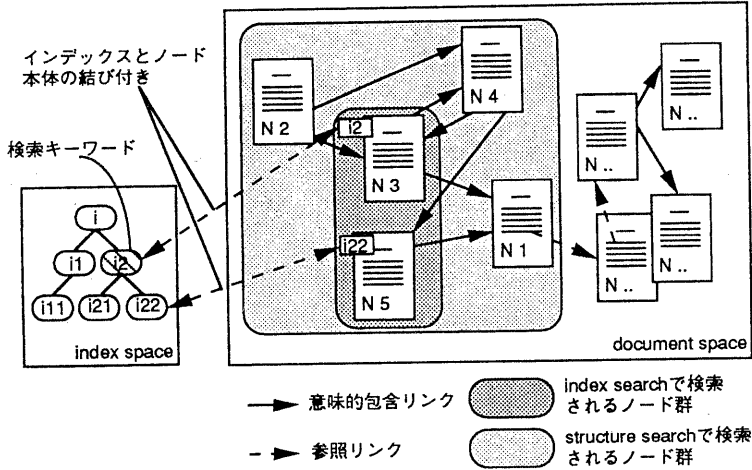


図 2: インデックスサーチとストラクチャサーチ

次のようなブール演算を行う（ここで、 I は単位マトリクス）。

$$(A+I)^{k-1} \neq (A+I)^k = (A+I)^{k+1} = M \quad (2)$$

この結果、得られたマトリクス M (式3) はグラフ理論で可到達マトリクスと呼ばれるものである。もし、この可到達マトリクスの要素 $m_{ij} = 1$ であるならば、これは推移性によって i から j へ直接的および間接的に関係がおよぶことを意味している。

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

階層ネットワーク構造の生成: 可到達マトリクスから階層ネットワーク構造を導くアルゴリズムは以下の通りである。ここで、 $R(N_i)$ と $A(N_i)$ を次のように定義する。

$R(N_i)$: ノード N_i から到達可能なすべてのノードの集合

$A(N_i)$: ノード N_i に到達可能なすべてのノードの集合

この2つの集合は、次の $R(N_i) \cap A(N_i)$ を導くためのものである。

$R(N_i) \cap A(N_i)$: ノード N_i から到達可能で、かつノード N_i に到達可能なノードの集合

N_i	$R(N_i)$	$A(N_i)$	$R(N_i) \cap A(N_i)$
1	1	1 2 3 4 5	1
2	1 2 3 4 5	2	2
3	1 3 4 5	2 3 4	3 4
4	1 3 4 5	2 3 4	3 4
5	1	2 3 4 5	5

表 1: マトリクス M の $R(N_i)$, $A(N_i)$, $R(N_i) \cap A(N_i)$

つまり、 $R(N_i) \cap A(N_i)$ に含まれるノードはグラフ理論での強連結、またはグラフのなかでループ状態にあるものということができる。ここで式3の可到達マトリクスの各要素についてそれぞれ求めたものを、表1に示す。

ノードが強連結の状態にあるときは、それらのノードをひとつにまとめた集合 $R(N_i) \cap A(N_i)$ を1つのノードとして取り扱う。ここで例では N_3, N_4 をひとつまとめのノードとみなして $\overline{N_3, N_4}$ と書き表わす。

表1において、 $R(N_i) \cap A(N_i) = R(N_i)$ を満たす N_i の集合を求める。この集合は、これに属さないノードのどれにも到達できないノードの集合であり、方向性グラフの最も上の層に位置するノードということになる。この例では N_1 がこれに相当する。最も上層のノードが決ったならば可到達マトリクス M からそのノードに対応する行と列を取り除き（すなわち、表1から N_1 を取り除く）、上述と同様の手続きをとって次に位置する層のノードを求める。先ほどの例では N_1 を取り除いた後なので N_5 が次の層のノードであることがわかる。このような手順を繰り返すことにより、上層のノードから順に取り出すことができる。このようにして階層化したノードの間の直接的関係を式1の直接関係マトリクス

を見ながら結ばば階層的有向グラフを得ることができる。図3(b)にその例を示す。

階層ネットワーク構造の線形化: 階層化されたネットワーク構造は、その後、図3(c)のように線形化される。下層に位置するノードほど本の前の方のページに置かれ、上層に位置するノードほど後ろの方のページに置かれる。このようにネットワーク構造を線形化することで、ユーザが本をはじめの方から順番にバラバラ見ているときに、本に書かれている内容の概要をまず見て、その後、そのページのすぐ後ろ（あるいは数ページ後ろ）のページに書かれているより細かい記述を見ることになる。すなわち、ハイパーメディア中の一つのリンクが保持する情報は2つのノード間のミクロな関係であるが、OpenBook システムによって生成された本の中にはマクロな情報の流れが存在しており、ユーザがノード群を体系的に把握するための手助けをしているのである。

6 プロトタイプシステム

6.1 システム構成

プロトタイプシステムは「OpenBook マネージャ」と「OpenBook ウィンドウ」の2つのパートから構成されている。OpenBook マネージャはインデックススペースを管理する機能（たとえば、階層化インデックススペースの表示、構造の変更、インデックスの変更）と、一つあるいは複数の本（OpenBook ウィンドウ）を生成するための機能を提供している。一方、OpenBook ウィンドウは本を表示するためのウィンドウであり、プレーンテキスト、イメージ画、ポストスクリプト描画を表示する能力を持っている（プレーンテキスト、ポストスクリプト描画に関しては、それぞれ、テキストエディタ、*idraw*エディタを用いて随時画面上で編集可能）。

システムはリンク情報とノード情報を物理的に別のデータベースで管理しているが、それらは調和をとって管理されている。プロトタイプシステムでは、ノード本体を UNIX ファイルとして保存し、リンク情報をリレーショナルデータベース (RDB) に保存している。この RDB は、この他、インデックススペースを管理したり、ノードに付けられたインデックスを管理したり（ノード本体は UNIX ファイルに保存されるが、ノード識別子は RDB で管理される）、問い合わせ検索を支援するために用いられる。

プロトタイプシステムは SUN ワークステーション上で稼働しており、GNU C++ を用いて実装された。また、ユーザインタフェースの構築のために、ウィンドウ環境として OpenWindows 2.0、NeWS 2.1 を使い、さらに、XView、TNT-2.0 Toolkit を使用した。

6.2 OpenBook ウィンドウの GUI

図4は OpenBook ウィンドウのグラフィカルインタフェース (GUI) の一例を示している。図で表されているように、ウィンドウはあたかも普通の本のような外観を持っている。「紙・本メディア」の利点を継承しつつ、いかに電子化したメリットを融合してゆくかが本メタファを導入した場合の課題であったが、人間が使い慣れた本の感覚を維持するために可能な限り物理的な本に近い表現をとった。システムは、ユーザに対し、次に述べるシーケンシャル（線形）およびランダム（非線形）アクセス方法を提供している。

線形アクセス: 本の右（左）ページの任意の場所をマウスを用いてクリックすると、そのページがめくられ、一つ後ろ（一つ前）のページに進むことができる。また、クリックを素早く押せば、ページは高速にめくられ、本をバラバラとめくって情報を拾い読みしているような感覚で大量の情報にざっと目を通すことができる。このページめくりと同調して左右ページの厚みが動的に変更するため、ユーザは常に本の中での位置を把握できる。

非線形アクセス: ユーザは、本の厚み部分をクリックすることにより、たとえば「本のかかなり後ろの方」といった曖昧な記憶から、目的のページの近傍へジャンプできる。また、ハイパーメディアリンクのアンカー（参照/意味的包含リンクのアンカー）がページ内に存在していたならば、そのリンクをたどることもできる。この時、飛び先のノードが同じ本の中に存在していたならば、そのページへジャンプし、そうでなければ、本に含まれていないハイパーメディア中のノードにジャンプすることになる。

ディスプレイ上には同時に複数の本を開くことが可能であり、ユーザは、それぞれの本に書かれた情報を見比べながら作業を進めることができる。また、本の中のノードと、別の本に表示されているノードとの間にリンクを張ることも可能である。

さらに OpenBook ウィンドウを任意の大きさにリサイズすることができる。この時、リサイズしてもページ中の個々のオブジェクトのレイアウトはまったく変更されず、各オブジェクト自身がウィンドウのスケールに比例して拡大・縮小される（PostScript の *scale* 命令を使用）。リサイズによってレイアウトが変化してしまうと、ページ中のオブジェクトの位置を人間が空間的に記憶している場合、そのページを見つけ出せなくなってしまうためである。また、縮小されたウィンドウはアクティブ・アイコンとしても使用できる（図4参照）。

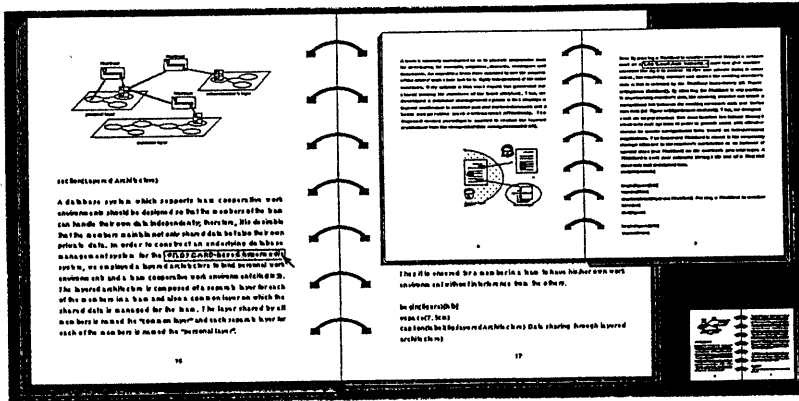


図 4: OpenBook ウィンドウ

7 おわりに

本稿では、迷子の問題 (disorientation) の解消と、ユーザの知覚上の負担 (cognitive overhead) の軽減を目指した共有ハイパーメディアシステム OpenBook を提案した。また、ナビゲーション検索および他の研究に見られる試みが抱える数々の問題点について言及した。これらの問題点に対して、著者らは、本メディアとハイパーメディアを統合することを考え、ハイパーメディアのブラウザの構築に本メタファを導入した。本というメディアは我々にとって非常になじみ深いものであり、本の物理的性質がもたらす数々の利点により、読者は情報を読み進めて行く上で大きな安心感を得ることができる。また、本形式のブラウジング手法は大量の情報にざっと目を通すことができるという利点を持っており、読者は本の内容の大筋を素早く把握し要点だけを拾い出すことができる。

さらに、問い合わせ検索機構と ISM ノード構造化法を組み合わせて使用することにより、ハイパーメディア中の個々のリンクが保持する2つのノード間のマイクロな関係から、ハイパーメディア中のマクロな情報の流れを抽出することを可能にし、ユーザがノード群を体系的に把握するための支援を可能にした。この ISM ノード構造化法は、階層化ネットワーク構造をグラフィカルマップに表示することで、効果的なノードマップを作成するためにも有効であろう。

問題となるのは、ストラクチャサーチによって本に取り込むノードの範囲をどの程度にするべきかという点である。これに対しては、ユーザの選択によって取り込む範囲を変更できる仕組みが必要であり、今後の課題となっている。

参考文献

- [1] Ichimura, S., Matsushita, Y., "A PilotCard-Based Hypermedia Network Integrated with a Lay-

ered Architecture-Based OODBMS and an Object-Forwarding Mail System", Proc. ACM Computer Science Conference, March 1992.

- [2] Utting, K., Yankelovich, N., "Context and Orientation in Hypermedia Networks", ACM Trans. on Information Systems, January 1989.
- [3] Bernstein, M., "The Bookmark and the Compass: Orientation Tools for Hypertext Users", ACM SIGOIS Bull., Oct. 1988.
- [4] Frisse, M. E., Cousins, S. B., "Information Retrieval From Hypertext: Update on the Dynamic Medical Handbook Project", Proc. Hypertext'89, 1989.
- [5] 荒井, 市村, 塚田, 松下, "グループワークのための電子ブックの提案", 情報処理学会第 43 回全国大会 5-77, 1991.
- [6] Halasz, F. G., "Reflections on Notecards: Seven Issues for the Next Generation of Hypermedia systems", Commun. ACM, July 1988.
- [7] Delisle, N., Schwartz, M., "Neptune: a Hypertext System for CAD Application", Proc. ACM SIGMOD, May 1986.
- [8] Thompson, R. H., Croft, W. B., "Support for Browsing in an Intelligent Text Retrieval System", Int. J. Man-Machine Studies, vol. 30, 1989.
- [9] Watters, C., Shepherd, M. A., "A Transient Hypergraph-Based Model for Data Access", ACM Trans. on Information Systems, April 1990.
- [10] Robertson, G., McGracken, D., Newell, A. "The ZOG Approach to Man-Machine Communication", Int. J. Man-Machine Studies, vol. 14, 1981.
- [11] Walker, J. H., "Supporting Document Development with Concordia", IEEE COMPUTER, January 1988.
- [12] 佐藤, "教材構造化支援", 電子情報通信学会誌 vol.71 No.4, 1988.